Public Key Infrastructure

Murat Kantarcioglu

What is PKI

- How to ensure the authenticity of public keys
- How can Alice be sure that Bob’s purported public key really is Bob’s key (not Charlie’s)
- A PKI is a secure system that is used to manage and control certificates
- Several aspects:
  - The system should function without the active intervention of the user.
  - Public key signature scheme is used
  - No need to keep prior shared secret keys between two parties

Components of PKI

- Certificate Issuance:
  - Most PKIs have one or more trusted authorities (certification authorities) that control the issuing of certificates.
  - Before a certificate issued the identity of the user must be verified.
  - A secure procedure is needed to generate and transmit the public key and private key to the user

Components of PKI

- Certificate Revocation:
  - Revoking certificate before a normal expiration date
  - When private key being lost or other fraudulent use of the key
  - Similar to credit cards stolen
  - Additional infrastructure is required to recognize revoked certificates

Components of PKI

- Key backup / recovery / update:
  - Secure storage of users' private keys by the administrator of the PKI, in case users lose or forget their private keys
  - User has to prove its identity before being allowed to access a stored private key
  - When a certificate is about to expire the old key can be used to transfer the new key
  - More efficient than generating new keys and certificates from scratch

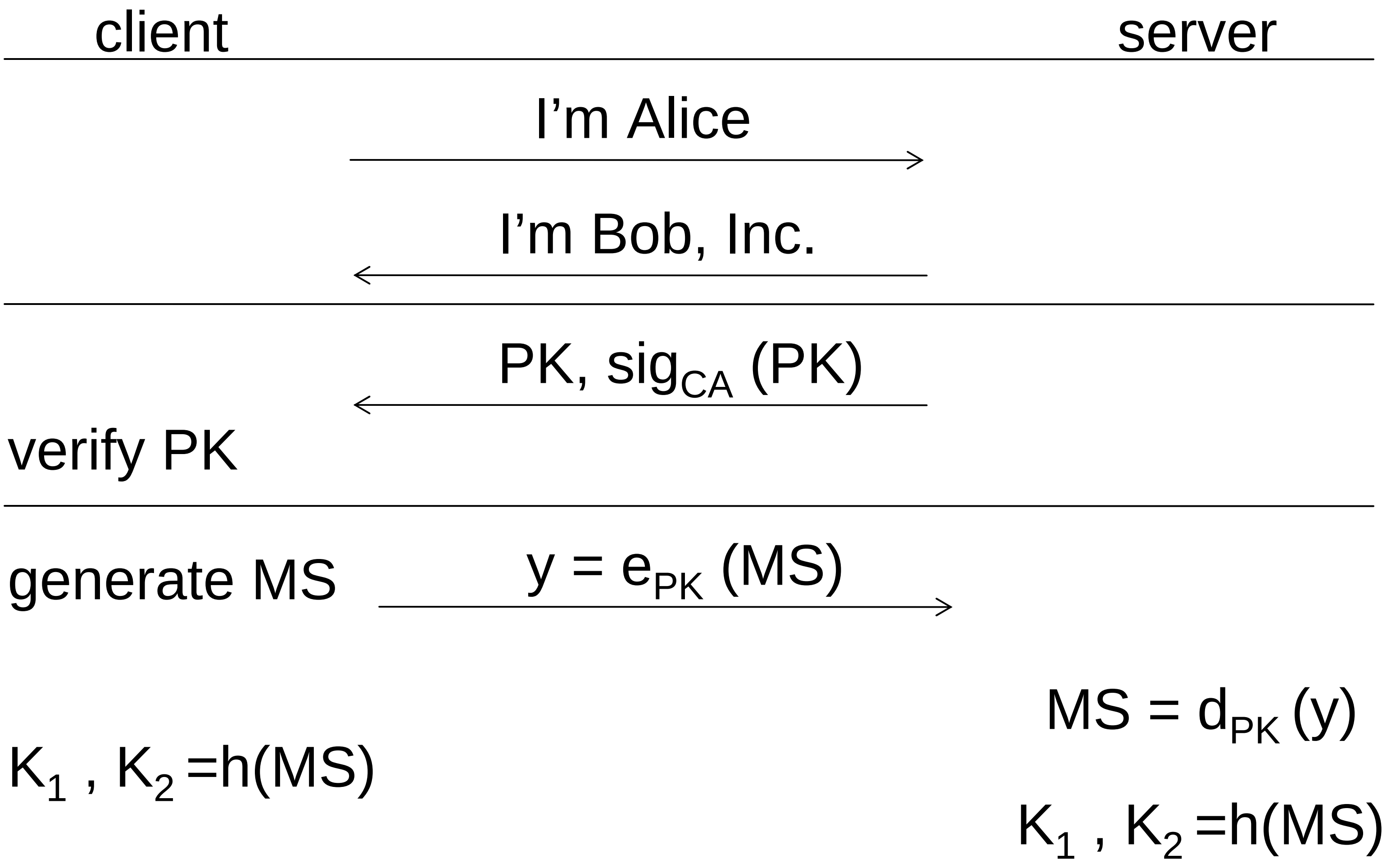
Components of PKI

- Timestamping:
  - Certificates usually have fixed length validity periods.

Secure Socket Layer

- SSL is used to facilitate online purchases from a company's web page using a web browser
- Only the server is required to supply a certificate during an SSL session
- The client may not even have a public key or certificate

Setting up an SSL Session



client

server

I’m Alice

I’m Bob, Inc.

PK, sig_{CA} (PK)

verify PK

generate MS

$y = e_{PK}$ (MS)

$MS = d_{PK}$ (y)

K_1 , K_2 =h(MS)

K_1 , K_2 =h(MS)

Certificates

- Certificates are building blocks of PKIs
- A certificate binds an identity to a public key
- Everyone has access to an authentic copy of the public key of the CA
- X.509 v3 certificates contain the following fields:
  - Version number
  - Serial number
  - Signature algorithm ID

Certificates

- X.509 v3 certificates contain the following fields:
  - Issuer name
  - Validity period
  - Subject name (i.e. the certificate owner)
  - The cert. owner public key
  - Optional fields
  - The CA’s signature on all the previous fields

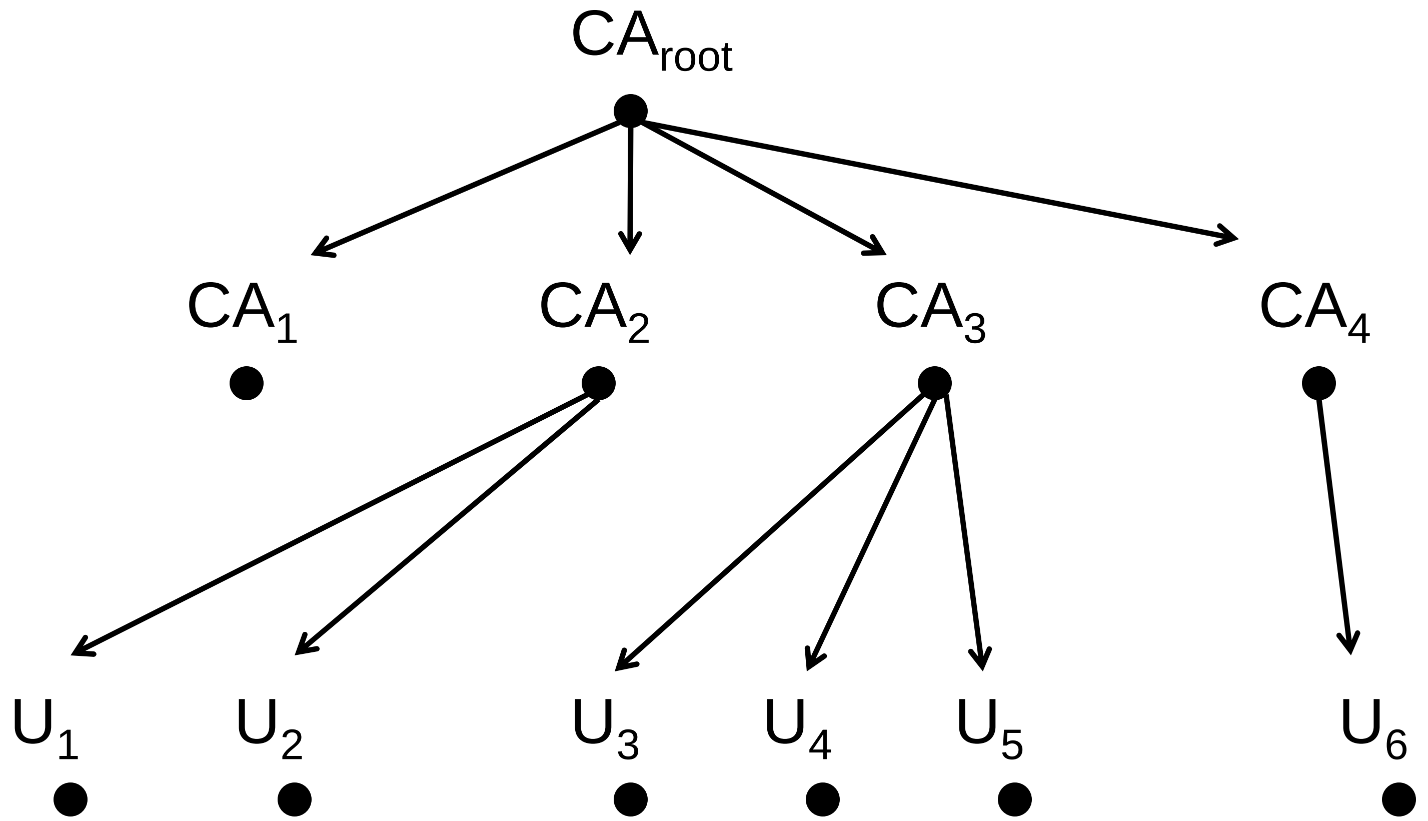
Trust Models

- Using certificate paths to sign a certificate
- Trust models:
  - Strict hierarchy
  - Networked PKIs
  - Web browser model
  - User centric model

1) Strict Hierarchy Model

- The root CA is called a trust anchor
- Root CA may issue certificates for lower level CAs
- Any CA can issue certificates for end users
- x → y means that the entity corresponding to node x has signed a certificate for the entity corresponding to node y.

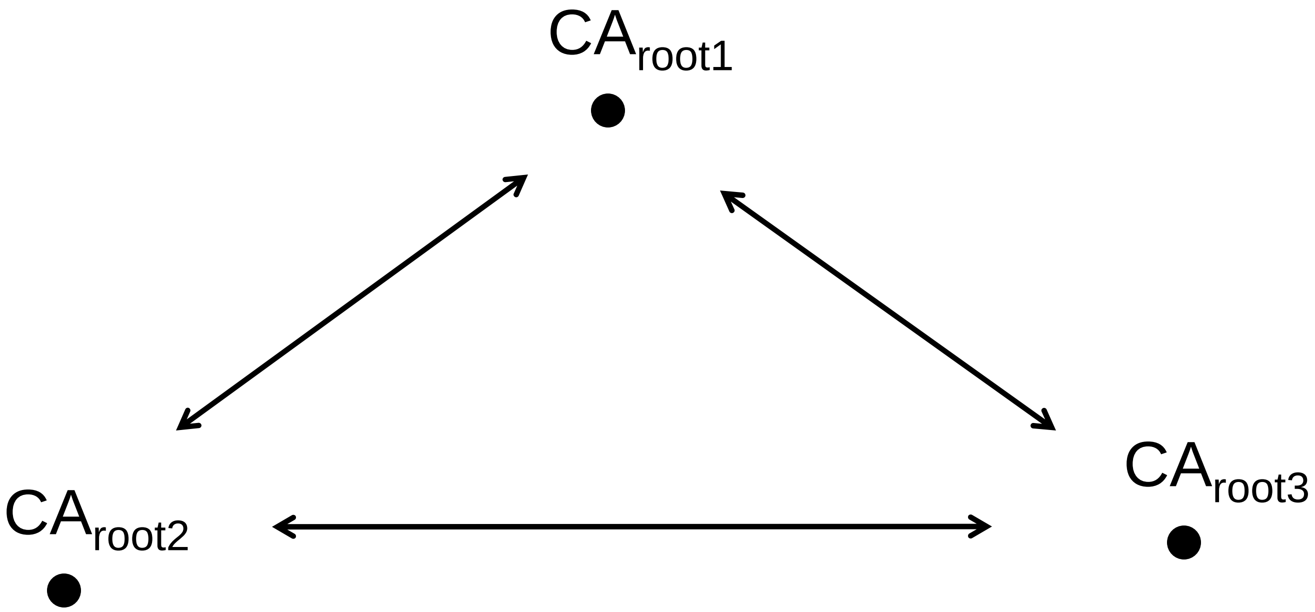
1) Strict Hierarchy Model (Cont.)



2) Networked PKIs

- Strict hierarchy may work well within a single organization.
- Sometimes it may be desirable to connect root CAs of two or more different PKI domains.

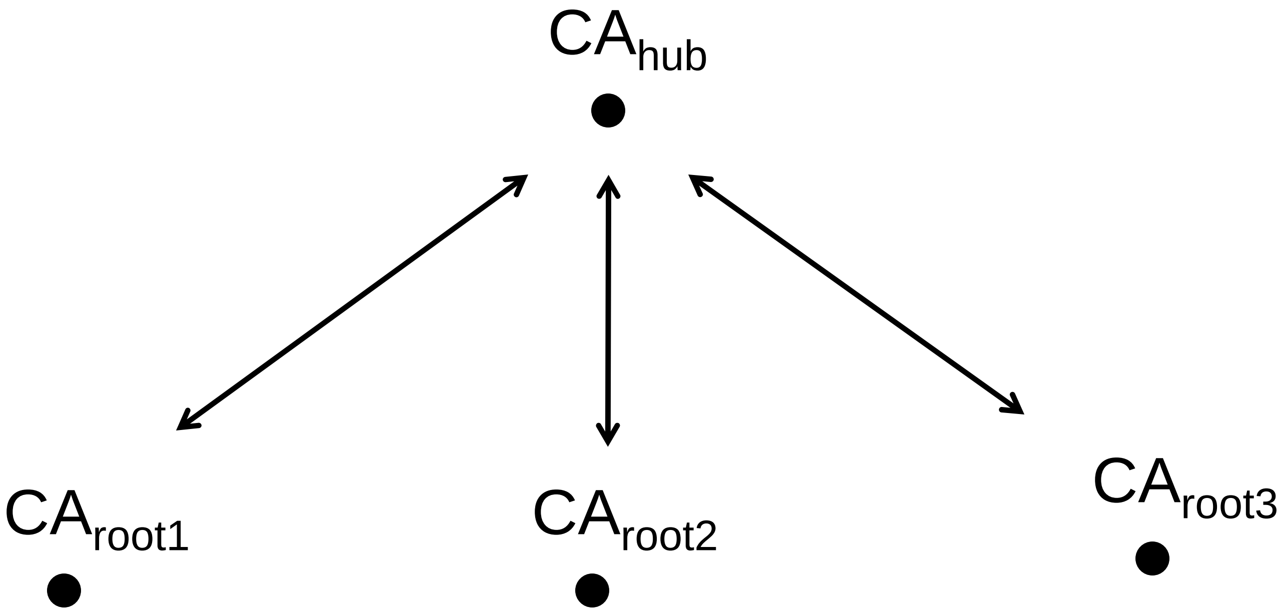
Networked PKIs – The Mesh Configuration



Cross certification is used among multiple CAs

n root CAs will require n(n-1) cross certifications

The Hub-and-spoke Configuration



n root CAs each cross-certify independently with a new hub CA.

2n cross certifications are required

3) The Web Browser Model

- Web browsers come preconfigured with a set of “independent” root CAs
- These are treated by the user of the browser as trust anchors.
- The provider of the browser is assumed to be trusted.
- Problems:
  - No information about the preconfigured CAs
  - Not enough expertise to edit the lists

3) The Web Browser Model (Cont.)

- Problems:
  - No mechanism to revoke a root CA
  - User may accept self signed certificates
  - Expiration dates

4) Pretty Good Privacy (PGP)

- Every user is his or her own CA.
- A PGP certificate contains an e-mail address (ID), a public key (PK) and one or more signatures on this (ID, PK) pair.
- Alice's self signed certificate:
  - Cert(Alice) : (data, signatures)

  where

  data : (ID = alice@utdallas.edu , PK = 12345)

  signatures : $\text{sig}_{\text{Alice}}$(data)

4) Pretty Good Privacy (PGP)

- Other users might also create signatures on the data on Alice’s certificate.
- signatures : (sig$_{\text{Alice}}$(data), sig$_{\text{Bob}}$(data))
- The signatures on a certificate help to verify the certificate’s authenticity to other users
- Collection of certificates: keyring
- Associated with each certificate in the keyring is an *owner trust field* (OTF) and a *key legitimacy field* (KLF)

The Future of PKI

- Many potential difficulties associated with large-scale deployments of PKIs
- Who should be responsible for development, maintenance and regulation of PKIs? Governments? Industry?
- What standards should be used in PKIs?
- Interoperability problems
- Different PKI needs in different environments